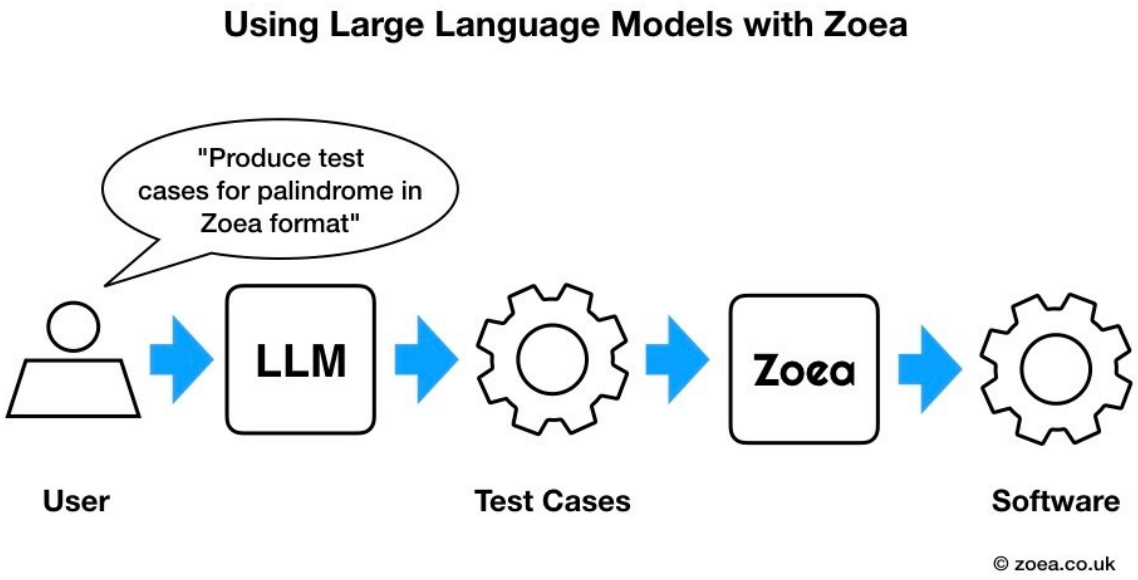
Zoea

Research note 26

Using Large Language Models with Zoea

Edward McDaid & Sarah McDaid
16 May 2023

Large language models show promise in automating the production of software however significant hurdles remain. Plugging Zoea into the workflow makes a lot of these problems go way.



It would seem that software generation is finally a thing. Large language models (LLMs) have taken the world by storm in recent months and have shown potential in many domains. This includes the ability to generate software - at least to some extent. In itself, this is all very impressive, however, another important change is in the perception of AI by people and business. There is a palpable realisation that this technology will quickly become part of everyone's lives. This includes a widespread and growing acceptance that code generation is the near future of software development.

At the same time it is clear that we're not there yet. Amongst the near total news saturation around LLMs there are many reports of problems associated with generated code. These run the full gamete from functional errors to security and performance issues, and of course bugs. LLMs will sometimes confidently assert incorrect information in a little understood phenomenon dubbed 'hallucination'. As a result there remains a fair bit of uncertainty and risk associated with LLM-generated code.

A more insidious problem is that users will often simply not understand (or even bother to understand) the code that they are presented with. It is a good thing to democratise software development by enabling more people to code without having to learn a programming language. However, simply providing source code as an output isn't helpful for many. This problem becomes more acute as the size of the code produced increases. Perhaps the right process for software generation is a little more involved than simply "write fast Fourier transform in Cobol".

There are also some more fundamental problems associated with LLM-generated code. Transparency is a major problem for all approaches based on deep learning and neural networks. This means that it is not easy to understand how the AI came up with a particular output. There are also a lot of concerns on the part of open source software developers that their work has been used in training an AI, might somehow be stored verbatim in its model and can be reproduced more or less exactly without attribution. Such problems were anticipated years ago but now sadly won't be settled until they have been through the courts.

When it comes to code generation LLMs aren't the only game in town. Zoea is another type of AI technology that generates code directly from a set of test cases. This makes it easy to learn and also easy to use for technical and non-technical people alike. Unlike LLMs, Zoea can guarantee that all of the code it produces works and meets the functional requirements. It is also capable of explaining its reasoning and infringes nobodies intellectual property.

Zoea is not in competition with LLMs but is rather a complementary approach. We share the vision of a future where coding is - at least mostly - a thing that happens within machines. Both approaches have some benefits but what happens if you plug them together?

LLMs can produce code in many languages but they can also produce test cases. Google Bard for example clearly has some passing familiarity with Zoea and can currently produce Zoea test cases in YAML format when requested. As test cases go these aren't bad and they will frequently produce the required code when fed into Zoea. Currently it is a manual step to input the test cases into Zoea but it is an easy matter to automate this on either end through an API.

In effect we can combine an LLM with Zoea to produce a composite system that delivers the benefits of both approaches. LLMs are able to leverage their general and coding knowledge to produce good test cases and make test case production trivial. Users can understand the requirements expressed in the test cases. Zoea produces surprise free code that delivers the required functionality. What's not to love?

Learn more at **zoea.co.uk**